CLAIMS:

What is claimed is:

1.   A method for full speculation of instruction
processing in a multiprocessor data processing system
comprising:

       issuing from a processor a barrier operation on a
system bus of said data processing system; and

       executing operations associated with instructions
sequentially following said barrier operation in an
instruction sequence prior to completion of said barrier
operation.

2.   The method of Claim 1, wherein said executing step
executes said operations, prior to said issuing step.

3.   The method of Claim 1, wherein said executing step
further comprises:

       issuing a load request for data;

       responsive to a return of said data, immediately
forwarding said data to a register of said processor; and

       providing said data to subsequent processes that
utilize said data.

4.   The method of claim 3, further comprising setting a
flag within said register when said barrier operation has
not yet completed, wherein said flag indicates that each
instruction executed and each result generated by said
subsequent processes and stored within said register is
speculative, pending a completion of said barrier
operation.

5.   The method of Claim 4, further comprising:

monitoring for said completion of said barrier
operation;

responsive to said completion, resetting said flag
and concurrently indicating said register as non-
speculative.

6.   The method of Claim 5, wherein further, when an
invalidate is received prior to said completion, said
processor discards said data and each of said result from
said register.

7.   The method of Claim 6, wherein said operations
include load requests and branch instructions, and
wherein further said method provides embedded branch
speculation within said operations and speculative load
request issuing within a branch path.

8.   A multiprocessor computer system comprising:

a plurality of processors interconnected by a system
bus, wherein said processors including a first processor
that speculatively issues load requests and processes
subsequent instructions utilizing data returned by said
load request before a completion of a barrier operation
that is sequentially before said load requests and
subsequent instructions in an instruction sequence; and

a memory hierarchy connected to said plurality of
processors via said system bus that sources said data.

9.   The multiprocessor computer system of Claim 8,
wherein said first processor comprises a load/store unit
with logic that controls issuing of load and store
instructions before completion of a preceding barrier
operation to provide said data to a register of said
first processor prior to a return of an acknowledgment
for said preceding barrier operations.

10.   The multiprocessor computer system of claim 8,
wherein said first processor further comprises:

execution units that processes instructions that
utilize said data when said data is placed in said
register; and

logic, affiliated with said register, that sets a
flag within said register when a value resulting from

8 executing said instructions is placed in said register

9 prior to said completion, wherein said flag messages to

10 the execution units that said instruction and said

11 results are speculative, pending a completion of said

12 barrier operation.

1 11. The multiprocessor computer system of claim 8,

2 wherein said logic further resets said flag responsive to

3 said completion.

1 12. The multiprocessor computer system of claim 11,

2 wherein said first processor further comprises a

3 plurality of execution queues and logic for setting a bit

4 associated with an entry of said queues to indicate

5 whether an instruction placed in said entry is

6 speculative with respect to said barrier operation.

1 13. The multiprocessor computer system of claim 11,

2 wherein said first processor further comprises a

3 plurality of execution queues and logic for setting a bit

4 associated with an entry of said queues to indicate

5 whether an instruction placed in said entry is

6 speculative with respect to an unresolved branch

7 instruction that precedes said instruction in said

8 instruction sequence.

14. A processor comprising:

a plurality of execution units including a
load/store unit, wherein said load/store unit
speculatively executes load requests and offer other
execution into speculative execute other instructions
before completion of a barrier operation that precedes
said load requests and other instructions in an
instruction sequence;

a rename register that includes a plurality of
entries, wherein each entry has a speculation flag and an
associated general purpose register identifier; and

logic for setting said speculation flag to indicate
when a value stored in said entry is speculative, pending
completion of said barrier operation.

15. The processor of Claim 14, wherein said load/store
unit provides data returned by said load requests
immediately to an entry of said rename register for
utilization within subsequent processes that require said
data.

16. The processor of Claim 15, wherein said load/store
unit messages said execution units and said logic when
said barrier operation completes.

17.   The processor of Claim 16, wherein, said logic, responsive to a receipt of a message indicating successful completion of said barrier operation, resets each flag associated with a register entry that was speculative with respect to said barrier operation.

18.   The processor of Claim 17, further comprising:

a plurality of issue queues associated with said execution units in which instructions to be executed are placed; and

logic for indicating that a particular instruction within one of said issue queues is speculative with respect to the barrier operation.

19.   The processor of Claim 17, further comprising:

a plurality of issue queues associated with said execution units in which instructions to be executed are placed; and

logic for indicating that a particular instruction within one of said issue queues is speculative with respect to an unresolved branch instruction that precedes said instruction within said instruction sequence.

20.   The processor of Claim 18, further comprising:

an enhanced internal instruction set architecture that includes a setable bit, which indicates whether an instruction is speculative, wherein said logic sets said setable bit responsive to whether said barrier operation has completed; and

when said barrier operation has completed, said logic resets said bit.

21.   The processor of Claim 18, wherein said issue queues includes a speculation bit associated with each entry location, wherein said speculation bit is set by said logic when said particular instruction is placed in an associated entry location, and reset only when said barrier operation has successfully completed.

22. A data processing system comprising:

a memory;

at least two processors interconnected to each other
and said memory via a system bus, wherein a first
processor comprises:

a plurality of execution units including a
load/store unit, wherein said load/store unit
speculatively executes load requests and offer other
execution into speculative execute other
instructions before completion of a barrier
operation that precedes said load requests and other
instructions in an instruction sequence;

a rename register that includes a plurality of
entries, wherein each entry has a speculation flag
and an associated general purpose register
identifier; and

logic for setting said speculation flag to
indicate when a value stored in said entry is
speculative, pending completion of said barrier
operation.

23. The data processing system of Claim 22, wherein said
load/store unit provides data returned by said load
requests immediately to an execution unit of said
processor for utilization within subsequent processes
that require said data.

24. The data processing system of Claim 23, wherein said
load/store unit messages said execution units and said
logic when said barrier operation completes.

25. The data processing system of Claim 24, wherein,
said logic, responsive to a receipt of a message
indicating successful completion of said barrier
operation, resets each flag associated with a register
entry that was speculative with respect to said barrier
operation.

26. The data processing system of Claim 25, further
comprising:

    a plurality of issue queues associated with said
execution units in which instructions to be executed are
placed; and

    logic for indicating that a particular instruction
within one of said issue queues is speculative with
respect to the barrier operation.

27. The data processing system of Claim 26, further
comprising:

    an enhanced internal instruction set architecture
that includes a setable bit, which indicates whether an

instruction is speculative, wherein said logic sets said

setable bit responsive to whether said barrier operation

has completed; and

when said barrier operation has completed, said

logic resets said bit.

28.   The data processing system of Claim 26, wherein said

issue queues includes a speculation bit associated with

each entry location, wherein said speculation bit is set

by said logic when said particular instruction is placed

in an associated entry location, and reset only when said

barrier operation has successfully completed.